In re Appln. Of: Sarat C. Manni et al.
Application No.: 09/867,303

REMARKS

Claims 1-26 are currently pending in the application. Claims 1, 3-6, 8-10, and 12 have been amended as indicated hereinabove to correct minor grammatical, typographical, and antecedent basis issues. However, none of these amendments affect the scope of the claim or the full scope of equivalents that should be afforded these claims as these amendments are not made for any reason related to patentability of these claims in view of the prior art. A replacement specification including paragraph numbering and the correction of a typographical error accompanies this amendment. No new matter has been added in this replacement specification. Claims 1-26 remain pending in this application, stand rejected, and are at issue herein. Reconsideration of claims 1-26 in view of the foregoing amendments and following remarks and indication of the allowability thereof at an early date are respectfully solicited.

The Examiner has objected to claims 1 and 10 for grammatical and typographical informalities. Reconsideration of this ground of objection in view of the foregoing amendments to claims 1 and 10 is respectfully solicited.

The Examiner has rejected claims 3-13 under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, the Examiner indicated that the designation of various buttons in these claims lack antecedent basis. While the applicants' respectfully traverse this ground of rejection as they believe the designation of the various buttons in the claims do particularly point out and distinctly claim the subject matter which the applicants regard as the invention, the applicants have nonetheless complied with the Examiner's suggestion to change the label of these various buttons. However, the applicants' respectfully submit that such a ministerial change does not affect the scope of these claims in any way related to patentability, and therefore respectfully submit that these claims are still subject to the full scope of protection and equivalents as if they had originally been filed in their amended form.

The Examiner has rejected claims 1-26 under 35 U.S.C. §103(a) as being unpatentable over "Universal Plug and Play Device Architecture" document in view of U.S. Patent No. 6,272,537 to Kekic et al. The applicants' have thoroughly considered this architecture document and the '537 patent, as well as the Examiner's combination and application thereof, but must

respectfully traverse this ground of rejection in view of the following remarks. Reconsideration of this ground of rejection and indication of the allowability of claims 1-26 at an early date are therefore respectfully solicited.

The Examiner correctly recognizes that the UPnP architecture document does not provide any teaching or suggestion of a generic user control point tool as recited in claim 1. This recognition by the Examiner confirms the applicants' description of the UPnP architecture in the background section of the originally filed application. The Examiner also recognizes that Kekic '537 also does not teach a generic user control point tool. However, the Examiner indicates that the "auto discovery panel" shown in Fig. 27 of Kekic '537 and the "MIB browser" shown in Fig. 36 of the Kekic '537 patent "are considered to constitute a generic user control point tool for discovering, controlling, and displaying network devices."

It is axiomatic in the patent law that to establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art references when combined must teach or suggest all of the claim limitations. Importantly, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicants' disclosure.

When these requirements are considered in assessing the combination of these references, the applicants' respectfully submit that they are not met. Specifically, the applicants' respectfully submit that there is no suggestion or motivation to make the proposed combination, there is no reasonable expectation of success that the proposed combination would operate to discover, control, and display UPnP devices, and finally the combined references do not teach each and every limitation required by the claims of the present application.

The Examiner has indicated that the motivation to combine these references is that such combination "would have been advantageous to one of ordinary skill to utilize these combined teachings of the UPnP document and Kekic, because such a discovery panel, MIB Browser, and Navigation Tree provides a user interface for discovering and controlling network devices, even if the vendor of the network device does not provide an interface to control the device, as is

demonstrated by Keikic." However, the UPnP document itself already describes the existence of a control point that is able to discover and control UPnP devices on the network, even if the vendor of the network device does not provide an interface. As such, one of ordinary skill in the art would have no motivation to seek to modify the UPnP architecture by using a different discovery panel, MIB Browser and Navigation Tree provided by Kekic as such would be, at best, redundant of the control point that already exists and is defined by the UPnP architecture.

The Examiner has also stated that such combination may be motivated to discover and control network devices if the vendor of the network device does not provide an interface to control the device. However, the UPnP architecture already provides this functionality through the control point. The provision of a separate vendor generated user interface is optional in the UPnP architecture, and the UPnP control point is still able to detect and control UPnP devices for which the vendor has not generated a device specific user interface. As such, the applicants respectfully submit that one of ordinary skill in the art would not be motivated to modify the control point defined in the UPnP architecture because this control point does not suffer from any of the problems stated by the Examiner to support the proposed combination of references.

The problem with the control point defined in the UPnP architecture document is not that it is unable to discover and control UPnP devices, but that the currently defined mechanism does not provide a common user experience across all UPnP devices. This significantly complicates the user experience, and detracts from the promise of ease of use envisioned by UPnP. The combination of Kekic '537 does not satisfy this requirement. This is because Kekic '537 also provides various user interfaces to control different network devices. Specifically, only if the network element that has been discovered cannot be associated with an element manager will the MIB Browser be used to manage the element. This still results in multiple, different user experiences depending on the network element currently selected. That is, those network elements that can be associated with an element manager will utilize that user interface to control the element, while those that cannot be associated with an element manager will be managed via the MIB Browser.

As with the UPnP architecture, the system of Kekic '537 allows for multiple user interfaces that will not provide a common user experience across all devices. That is, the system of Kekic '537 utilizes different element managers to manage different computer network

9

elements, and only uses the MIB Browser when no element manager exists for a particular computer network element. Such a system does not appear to the applicants to be much different than the UPnP framework wherein vendor generated user interfaces are used to control those UPnP devices for which they exist, but utilizes the control point to control other UPnP devices that do not have vendor generated user interfaces.

In view of the above, the applicants' respectfully submit that there is no suggestion or motivation to combine these references. As such, no *prima facie* case of obviousness has been established. Reconsideration of this ground of rejection and indication of the allowability of claims 1-26 at an early date are respectfully solicited.

In addition to the foregoing, the applicants respectfully submit that there is not a reasonable likelihood of success that the proposed combination will operate. That is, the Examiner has proposed utilizing the discovery panel and MIB Browser to provide the user interface for discovering and controlling network devices in the UPnP architecture. However, in the UPnP architecture, UPnP specific protocols are utilized, including Simple Service Discovery Protocol (SSDP), General Event Notification Architecture (GENA), and Simple Object Access Protocol (SOAP). In the MIB Browser of Kekic '537, however, the network management protocol is the Simple Network Management Protocol (SNMP). As such, the network management data in the system of Kekic et al. '537 are MIBs. In Kekic et al. '537, then, it is appropriate to utilize a MIB Browser for those network elements that the discovery engine is unable to associate with an element manager. However, a MIB Browser may not be appropriate for use in the UPnP architecture in view of the different protocols used therein. As such, the applicants' respectfully submits there is no reasonable likelihood of success that these two desperate systems utilizing different protocols would succeed.

Finally, the applicants respectfully submit that even if the teachings of each of these references were used, they would still fail to teach each and every limitation required by the claims of the application. In discussing independent claim 1, the Examiner points to section 1.2.2 of the UPnP architecture document. However, this section merely describes the message format of the search request with M-Search. No display device is described whatsoever. When analyzing the MIB Browser illustrated in Fig. 36 of Kekic et al. '537, it is noted that this user interface is devoid of any field for displaying discovery options. Indeed, the MIB Browser of

10

Kekic et al. '537 is also devoid of any area for initiating a discovery process for UPnP devices. Instead, the MIB Browser is not utilized until a network device <u>has already been discovered</u> but cannot be associated with an element manager. As such, the applicants' respectfully submit that these two references taken in combination do not meet all of the limitations of independent claim 1. Therefore, the applicants' respectfully submit that independent claim 1, and those claims dependent thereon, are not rendered obvious by a combination of these references.

Dependent claim 5 requires a second button for viewing a presentation page for the UPnP device. However, the Kekic et al. '537 reference describes that the MIB Browser is only used if there is no element manager associated with a particular network element. As such, utilizing the MIB Browser as suggested by the Examiner is contrary to the teachings of claim 5 wherein a button is provided for opening a browser and connecting to a presentation URL for the presentation page.

With regard to independent method claim 14, the Examiner has indicated that the "limited search" field of Kekic et al. '537 teaches the step of receiving a discover type selection signal indicative of a user selecting one of a plurality of discovery types. However, the description of this "limited search" field in column 44, lines 53-57 clearly describes this as defining the scope of the search, not providing different discover types as defined in the originally filed specification. That is, regardless of whether the scope of the search is limited or not, all SNMP-enabled elements will be discovered. No limitation on the discover type is available. The limitation on the scope merely defines where the SNMP enabled elements must be connected for them to be discovered. *See* Kekic et al. '537, column 43, line 56 – column 44, line 51. As such, the applicants' respectfully submit that these method claims are also not rendered obvious by the combination of these references.

In view of the above, the applicants' respectfully submit that a *prima facie* case of obviousness has not been made for rejecting the claims of the present application. Therefore, the applicants respectfully submit that claims 1-26 are in condition for allowance. Reconsideration of claims 1-26 and indication of their allowability in view of the foregoing amendments and remarks are therefore respectfully solicited.

In re Appln. Of: Sarat C. Manni et al.
Application No.: 09/867,303

If the Examiner believes that a telephonic conversation will aid in the resolution of any issues not resolved herein, the Examiner is invited to contact the applicants' attorney at the telephone number listed below.

Respectfully submitted,

Jeffery J. Makeever, Reg. No. 37390
LEYDIG, VOIT & MAYER, LTD.
6815 Weaver Road, Suite 300
Rockford, Illinois 61114-8018
(815) 963-7661 (telephone)
(815) 963-7664 (facsimile)

Date: August 6, 2004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Sarat C. Manni et al.

Application No.: 09/867,303

Filed: May 29, 2001

For: GENERIC USER CONTROL POINT
         TOOL FOR UNIVERSAL PLUG AND
         PLAY (UPnP) DEVICES

Art Unit: 2173

Examiner: Blaine Basom

REPLACEMENT SPECIFICATION

(MARKED UP VERSION)

RECEIVED

AUG 1 3 2004

Technology Center 2100

GENERIC USER CONTROL POINT TOOL FOR UNIVERSAL PLUG AND PLAY (UPnP) DEVICES

RELATED APPLICATION

[0001]   This application is related to US Provisional Application Serial Number 60/226,989, filed August 22, 2000, the teachings and disclosure of which are hereby incorporated in their entirety by reference thereto.

TECHNICAL FIELD

[0002]   This invention relates generally to the discovery and control of Universal Plug and Play (UPnP) devices, and more particularly, to a generic tool capable of discovering, retrieving the description, viewing events, and controlling any UPnP device present in the network.

BACKGROUND OF THE INVENTION

[0003]   As described in the Universal Plug and Play Device Architecture document, the teachings and disclosure of which are hereby incorporated in their entirety by reference thereto, Universal Plug and Play (UPnP) is an architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices, and personal computers (PCs) of all form factors. It is designed to bring easy-to-use, flexible, standards-based connectivity to ad-hoc or unmanaged networks whether in the home, in a small business, public spaces, or attached to the Internet. Universal Plug and Play is a distributed, open networking architecture that leverages Transmission Control Protocol/Internet Protocol (TCP/IP) and the Web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and public spaces. UPnP is more than just a simple extension of the plug and play peripheral model. It is designed to support zero-configuration, "invisible" networking, and automatic discovery for a breadth of device categories from a wide range of vendors. This means a device can dynamically join a network, obtain an IP address, convey its capabilities, and learn about the presence and capabilities of other devices. Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) servers are optional and are used only if available on the network. Finally, a device can leave a network smoothly and automatically without leaving any unwanted state behind.

[0004]   The foundation for UPnP networking is IP addressing. Each device must have a DHCP client and search for a DHCP server when the device is first connected to the network. If a DHCP server is available, i.e., the network is managed, the device must use the IP addressed assigned to it. If no DHCP server is available, i.e., the network is unmanaged, the device must use Auto IP to get an address. In brief, Auto IP defines how a device intelligently chooses an IP address from a set of reserved addresses and is able to move easily between managed and unmanaged networks. If during the DHCP transaction, the device obtains a domain name, e.g., through a DNS server or via DNS forwarding, the device should use that name in subsequent network operations; otherwise, the device should use its IP address.

[0005]   Given an IP address, the first step in UPnP networking is called "Discovery." When a device is added to the network, the UPnP discovery protocol allows that device to advertise its services to control points on the network. Similarly, when a control point is added to the network, the UPnP discovery protocol allows that control point to search for devices of interest on the network. The fundamental exchange in both cases is a discovery message containing a few, essential specifics about the device or one of its services, e.g., its type, identifier, and a pointer to more detailed information. The UPnP discovery protocol is based on the Simple Service Discovery Protocol (SSDP).

[0006]   The second step in UPnP networking is called "Description." After a control point has discovered a device, the control point still knows very little about the device. For the control point to learn more about the device and its capabilities, or to interact with the device, the control point must retrieve the device's description from the Uniform Resource Locator (URL) provided by the device in the discovery message. Devices may contain other, logical devices, as well as functional units, or services. The UPnP description for a device is expressed in XML and includes vendor-specific, manufacturer information like the model name and number, serial number, manufacturer name, URLs to vendor-specific Web sites, etc. The description also includes a list of any embedded devices or services, as well as URLs for control, eventing, and presentation. For each service, the description includes a list of the commands, or actions, the service responds to, and parameters, or arguments, for each action. The description for a service also includes a list of variables. These variables model the state of the service at run time, and are described in terms of their data type, range, and event characteristics.

[0007]   The third step in UPnP networking is called "control." After a control point has retrieved a description of the device, the control point can send actions to a device's service. To do this, a control point sends a suitable control message to the control URL for the service (provided in the device description). Control messages are also expressed in XML using the Simple Object Access Protocol (SOAP) and transmitted usually through HTTP. Like function calls, in response to the control message, the service returns any action-specific values. The effects of the action, if any, are modeled by changes in the variables that describe the run-time state of the service.

[0008]   The fourth step in UPnP networking is called "eventing." A UPnP description for a service includes a list of actions the service responds to and a list of variables that model the state of the service at run time. The service publishes updates when these variables change, and a control point may subscribe to receive this information. The service publishes updates by sending event messages. Event messages contain the names of one of more state variables and the current value of those variables. These messages are also expressed in XML and formatted using the General Event Notification Architecture (GENA). A special initial event message is sent when a control point first subscribes; this event message contains the names and values for all evented variables and allows the subscriber to initialize its model of the state of the service. To support scenarios with multiple control points, eventing is designed to keep all control points equally informed about the effects of any action. Therefore, all subscribers are sent all event messages, during the period of subscription subscribers receive event messages for all evented variables that have changed, and event messages are sent no matter why the state variable changed (either in response to a requested action or because the state the service is modeling changed).

[0009]   The fifth and final step defined for UPnP networking is called "presentation." If a device has a URL for presentation, then the control point can retrieve a page from this URL, load the page into a browser, and depending on the capabilities of the page, allow a user to control the device and/or view device status. The degree to which each of these can be accomplished depends on the specific capabilities of the presentation page and device.

[0010]   Unfortunately, while this networking framework provides an adequate mechanism for the discovery, control, and presentation of UPnP devices, it does not provide a common user experience across all UPnP devices. That is, since each

individual UPnP device manufacturer develops its own URL page for presentation and control, the user experience for each device will be different. Further, since each manufacturer's URL page is designed for only that particular UPnP device, there is no common mechanism for discovering and controlling all UPnP devices on a network. As a result, a user at a control point computer must make multiple connections with a browser to multiple presentation pages to view and control all of the UPnP devices available on the network. This significantly complicates the user experience, and detracts from the promise of ease of use envisioned by UPnP.

SUMMARY OF THE INVENTION

[0011]   The instant invention discloses a generic UPnP tool (Generic User Control Point (UCP) Tool) that allows a user to discover, monitor, and control UPnP devices from various manufacturers on a network. In this way, the Generic UCP of the present invention presents a common user experience to UPnP networking heretofore unavailable across various manufacturers. The tool is preferably a Visual Basic (VB) application that uses the UPnP application programming interfaces (API's) in Windows Millennium (WinMe) and the Windows family of operating systems, although other implementations (e.g., c++, etc.) and operation with other operating systems are also envisioned.

[0012]   In one embodiment, the usage of the tool is split into two phases namely, discovering the devices and controlling them. The first phase involves choosing the type of search needed and discovering the devices. Once the UPnP devices have been discovered, the user may then browse the properties of the devices found. The second phase involves choosing a particular service of a selected device and then controlling it. During this stage, the user may also watch the eventing happen for the evented state variables for that particular service. In a preferred embodiment, the user may also utilize the tool to perform a query on any variable of the selected service in the device and control the service through a known list of actions that are specified in the service schema.

[0013]   The tool's user interface (UI) allows the user to choose a type of search to discover UPnP devices, either by type, by unique device name (UDN), or asynchronously, which is particularly useful when one wants to find all devices. Once the discovery process is started, the tool collects the information returned from the UPnP API in a pull-down list. A user may then select one of the UPnP devices from this pull-down list.

Once selected, the tool provides the user with the option of displaying the properties of the device, including the services provided thereby. A service of the device may then be selected from a pull-down list (if more than one is available) on the tool. Once a particular service has been selected, the user may then undertake to invoke an action of the service by entering the action and any required arguments. The eventing of the device may be monitored by viewing an Events field on the tool. The tool also provides the ability to open the manufacturer's presentation URL for the device if the user so desires.

[0014] Additional features and advantages of the invention will be made apparent from the following detailed description of illustrative embodiments, which proceeds with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] While the appended claims set forth the features of the present invention with particularity, the invention, together with its objects and advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

[0016] Figure 1 is a block diagram generally illustrating an exemplary computer system on which the present invention resides;

[0017] Figure 2 is a screen shot illustration of a user interface (UI) of an embodiment of the generic user control point (UCP) of the present invention;

[0018] Figure 3 is a screen shot illustration of a user interface (UI) display of a device properties page constructed in accordance with the teachings of the present invention; and

[0019] Figure 4 is a screen shot illustration of a user interface (UI) display of a service description viewer page constructed in accordance with the teachings of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0020] Turning to the drawings, wherein like reference numerals refer to like elements, the invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general

context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0021]   Figure 1 illustrates an example of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

[0022]   The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like. Additionally, and with particular application to the present invention, operation with UPnP devices in a networked environment with the computer 110 serving as a generic control point for the UPnP devices is also possible.

[0023]   The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed

by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0024] With reference to Figure 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Associate (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0025] Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF,

infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer readable media.

[0026]   The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Figure 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

[0027]   The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, Figure 1 illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0028]   The drives and their associated computer storage media discussed above and illustrated in Figure 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In Figure 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers hereto illustrate that, at a minimum, they are different copies. A user may enter

commands and information into the computer 20 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through a output peripheral interface 190.

[0029]   The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 110, although only a memory storage device 181 has been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0030]   When used in a LAN networking environment, the personal computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the personal computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Figure 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0031] In the description that follows, the invention will be described with reference to acts and symbolic representations of operations that are performed by one or more computer, unless indicated otherwise. As such, it will be understood that such acts and operations, which are at times referred to as being computer-executed, include the manipulation by the processing unit of the computer of electrical signals representing data in a structured form. This manipulation transforms the data or maintains it at locations in the memory system of the computer, which reconfigures or otherwise alters the operation of the computer in a manner well understood by those skilled in the art. The data structures where data is maintained are physical locations of the memory that have particular properties defined by the format of the data. However, while the invention is being described in the foregoing context, it is not meant to be limiting as those of skill in the art will appreciate that various of the acts and operation described hereinafter may also be implemented in hardware.

[0032] In one embodiment of the present invention operating in this environment, a generic user control point (UCP) for UPnP devices is presented as a user interface (UI) tool operable to (1) discover and (2) control UPnP devices that may be networked therewith. On such embodiment of an UI presenting the operability of the generic UCP tool of the present invention is illustrated in screen shot form in Figure 2 to which attention is now directed. This tool is referred to as a generic UCP because it is capable of discovering and controlling all UPnP compliant devices from a single UI, regardless of the UPnP device type and/or manufacturer. As such, it presents a single user experience for all UPnP devices, thus enhancing a user's ability to fully utilize the features promised by UPnP.

[0033] As may be seen from the exemplary generic UCP tool's UI 200 of Figure 2 (shown as a Windows Millennium application window), three discovery options are provided to allow a user to discover UPnP devices on a network. These options, which may be selected by a simple mouse selection, include "Find by Type" 202, "Find by UDN" 204, and "Async Find" 206. To enable the "Find by Type" 202 and the "Async Find" 206 selections, the tool 200 utilizes a file that the contains device information. Such a file may be named "devtype.txt" or other appropriate name. This file should contain the device information specified by UPnP and may be in the form:

```
upnp:rootdevice All Root Devices
urn:schemas-upnp-org:device.lighting.1 X10 Lighting Devices
```

A line in this file for each specific device type is added similarly as the second line above. The UPnP working group provides the definitions for the UPnP device schemas used in this file. To enable the "Find by UDN" 204 selection, the tool 200 utilizes a file that contains the UDN information for the UPnP devices. Such a file may be named "udn.txt" or other appropriate name. This file should contain the UDN information specified by the UPnP or the manufacturer, and may be in the form:

uuid:{7d50b574-4213-4b88-84d9-e5c9241fcb3a}

A line in this file for each specific device is added as above.

[0034] Once the type of discovery desired is selected by the user, the type of device to be found may be selected from a pull-down list in field 208, e.g. All Root Devices. For the Find by UDN option, the particular device UDNs will be listed in the pull down list for user selection. This pull down list is accessed in normal fashion via the arrow button 210. This arrow 210 is ghosted out for the Async Find option as all UPnP devices will be found using this option. The user may either do this or simply enter the data, e.g., upnp:rootdevice at the List Box below the buttons "Find by Type," "Find by UDN," and "Async Find."

[0035] Once the type of discovery and the type of devices to be discovered are selected, the user may select the Start Discovery button 212 to begin the discovery process by transmitting the standard UPnP multicast discovery message (for WAN connections, the Multicast support must be enabled). In response, all devices must respond if any of their embedded devices or services matches the search criteria in the discovery message (using the criteria selected above). All of the devices discovered are listed in a pull down list in the Devices Found field 214. This list is accessible via the pull down arrow button 216 in normal fashion. Once this process is complete, the first or discovery phase of operation is complete.

[0036] The second phase of operation for the Generic UCP tool of the present invention is the Control phase. In this phase, a user selects a device to be controlled, may view its properties, may choose a service to invoke or check the status of state variable of the device. To begin this phase, a user selects a particular device from the pull down list discussed above. Once a particular discovered device has been selected by the user, its properties may be displayed by selecting the Device Properties button 218. In one

embodiment of the invention, the device properties are displayed in a new Device Properties field 220, such as that illustrated in Figure 3. This field preferably includes all of the information provided by the UPnP device discovery process, including, e.g., the device's UDN, its friendly name, device type, model name and number, the model description, the model URL, the UPC and serial number, any presentation URL provided by the device manufacturer, the manufacturer itself, the manufacturer's URL. After the desired information has been obtained, the user may simply select the OK button 222 to return to the Generic UCP tool UI 200 (of Figure 2).

[0037] The services available for the selected device may be viewed in the Choose Service field 224 by selecting the pull down arrow button 226. The user may then select one of the services for the selected UPnP device from the pull down list. If information is available for the services of the device through, e.g., a service description document (SDD), the URL is provided in field 228 for the user's reference. If the user wishes to view the page, the View Service Desc. button 242 may be selected. Selection of this button 242 will open a separate window 256 illustrated in Figure 4 through which one can browse the SDD.

[0038] As may be seen from an examination of this Figure 4, the Service Description Viewer window 256 provides a listing of the SDD URL in field 258. Upon selection of the Go button 260, the SDD is retrieved and illustrated in display window 262. The exemplary SDD illustrated in display window 262 is presented in an expandable tree structure as is well known in the art. However, other display formats for the SDD may also be displayed. Through this Service Description Viewer window 256, the user may set selected actions via button 264 and variables via button 266 for the UPnP device. The user may also choose to populate the variable list via button 268 and the action list via button 270. To return to the Generic UCP tool window 200, the user may simply select the close button 272.

[0039] Returning now to Figure 2, if the user wants to query the value of any state variable of the selected service, the name of that variable may be selected from a pull down list of such variables in field 230. This pull down list is accessed by pull down arrow button 232. Once a state variable is selected, the Query button 244 may be selected to determine its value. The output of this query is returned to the Generic UCP tool of the present invention is presented in field 240. Likewise, if a user wants to invoke an action for the selected service, the action name may be selected from a pull down list of

available actions in field 234. This pull down list is accessed by pull down arrow button 236. Once an action is selected, its arguments are entered in field 238, and the Invoke button 246 may be selected to invoke that action. The results are shown in the field 240.

[0040]   The UPnP device's eventing information is shown in field 250. The Status field 252 provides status information for the Generic UCP tool itself to aid a user in the utilization of the tool. When the user is done using the tool of the present invention, the user may select the close button 248.

[0041]   The UCP tool also provides the user the opportunity to utilize the presentation UI (if any) designed by the UPnP device manufacture. If a device has a URL for presentation, then the UCP tool can retrieve a page from this URL, load the page into a browser, and depending on the capabilities of the page, allow a user to control the device and/or view device status. The degree to which each of these can be accomplished depends on the specific capabilities of the presentation page and device. This is accomplished in the exemplary embodiment of Figure 2 through the selection of the View Presentation button 254. Once this button is selected, a web browser, such as Internet Explorer, is opened by the UCP tool. The browser opens the presentation page from the manufacturer by directing the browser to the presentation page's URL. Full functionality provided by this presentation page is available to the user through the browser opened by the UCP tool.

[0042]   In view of the many possible embodiments to which the principles of this invention may be applied, it should be recognized that the embodiment described herein with respect to the drawing figures is meant to be illustrative only and should not be taken as limiting the scope of invention. For example, those of skill in the art will recognize that the elements of the illustrated embodiment shown in software may be implemented in hardware and vice versa or that the illustrated embodiment can be modified in arrangement and detail without departing from the spirit of the invention. Therefore, the invention as described herein contemplates all such embodiments as may come within the scope of the following claims and equivalents thereof.

ABSTRACT OF THE DISCLOSURE

A user control point tool allows generic discovery, control, and display of Universal Plug and Play devices from a common user interface. This generic UCP tool provides a common user experience for all UPnP devices, regardless of their individual manufacturers. The generic UCP tool allows discovery of UPnP devices by type, by unique device name, or asynchronously. The user may select one of the discovered devices, view its properties, and select one of the services provided for that device to control. Additional information from a service description document may be viewed, and a user may query the value of the state variables and invoke an action for a service for the selected UPnP device. The results of the action are displayed on the tool's UI, as is the eventing information for the UPnP device. Status information for operation of the generic UCP tool itself is also provided.